

IN THE SPECIFICATION

Please amend the specification as follows:

1. At page 2, lines 13-21:

One of the technologies underlying the World Wide Web is the prospect of using component software technology - the idea of breaking large, complex software applications into a series of pre-built and easily developed, understood, and changed software modules called components -- as a means to deliver software solutions much more quickly and at a lower cost (~~source: DCOM: A Business Overview, online at http://www.microsoft.com/ntserver/guide/dcom.asp~~). The goal is to achieve economies of scale for software deployment across the industry.

2. At page 31, line 3, through page 32, line 16:

The Distributed Component Object Model (DCOM) is a Microsoft model for distributed object computing. Within the DCOM environment, a remote DCOM Client Application 486 can make a request. The DCOM client 486 can be any type of client, including a Visual Basic client, a C++ client, or a Web Brower with Active

Server Pages (ASP). If the request made by the DCOM client 486 is a request for access to a remote process (interprocess request) the request is routed to proxy.dll [[404]]. Proxy.dll [[404]] is a code segment which receives any client requests targeted from a remote server, and will facilitate the necessary interprocess communications. The proxy.dll [[404]] understands how to communicate with the Client 486, and also understands how to communicate over an interface [[405]] which is shared by two or more processes. The proxy.dll [[404]] "marshals" the request parameters into an independent format so that they may be provided by the client process 486 over the COM-based interface [[405]], which conforms with the Microsoft DCOM Model. The stub.dll [[402]], which also understands how to communicate over the common interface [[405]], "un-marshals" the parameters into a format that can be understood by the DCOM Server Application.exe [[470]]. Thus, the DCOM environment allows machines with entirely different architectures (PCs, Workstations, etc.) to communicate using a common interface.

The specifics of the common interface are described in an Interface Definition Language (IDL) [[474]]. The IDL [[474]] is operated on by the Microsoft Interface Definition Language (MIDL) compiler [[476]] to create a set of .c [©] and .h (header) files [[470]]. Then, a second compiler ^{©++} [[400]] operates on the .c and .h files to create the stub.dll [[402]] and the proxy.dll

[[404]]. The proxy.dll [[404]] is linked to the DCOM Client Application.exe 486, and the stub.dll [[402]] is linked to the DCOM Server Application.exe [[470]].

Once the DCOM Server [[470]] un-marshals the parameters, the parameters are packaged into a single buffer and passed to the UDTXMO.dll [[460]]. The UDTXMO.dll [[460]] and the Ggate.exe [[466]], both of which are the subject to the present invention are the modules which "DCOM-enable" an OLTP enterprise server 452 such as the Unisys 2200 System.

This Page Blank (uspto)